Implicit Methods

(for solving differential equations)

Jungdam Won

Computer Science & Engineering

Seoul National Univ.

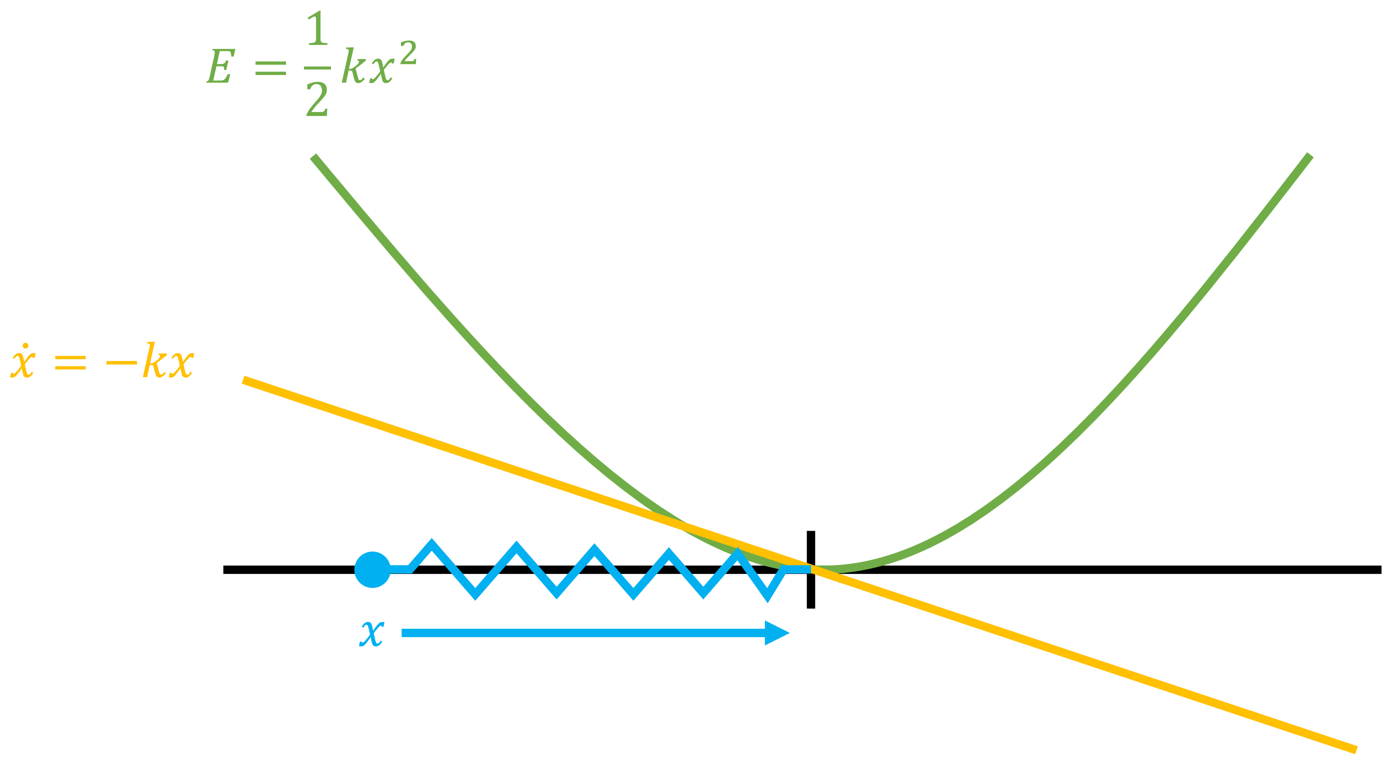
This material was created based on the slides and lecture notes of
Physically Based Modeling (SIGGRAPH 2001 course) by Andrew Witkin

Stability is Important

- If your simulation is unstable, you wouldn’t be able to get any meaningful result
  - If your step size is too big, your simulation will blow up
  - Sometimes you have to make the step size so small that you never get anyplace
- Nasty cases: cloth, constrained systems
- Solutions
  - **I. Use explosion-resistant methods**
  - II. Reformulate the problem

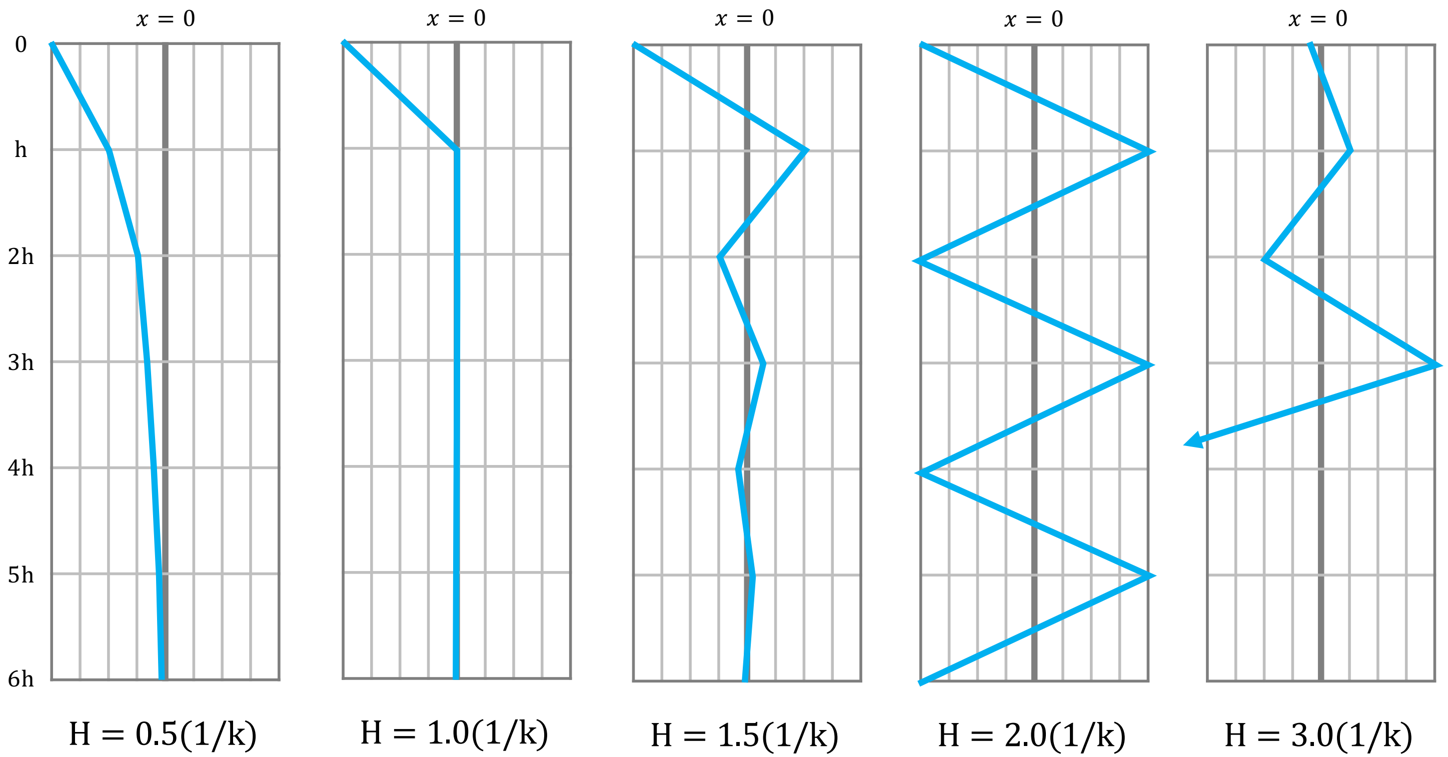
A Simple Equation

- A 1D particle governed by $\dot{x} = -kx$ where k is a stiffness constant



(Explicit) Euler Method Has a Speed Limit

$$\dot{x} = -kx \quad \Rightarrow \quad x_{t+1} = x_t + h\dot{x}_t = x_t - hkx_t$$



$h > \frac{1}{k}$: oscillate

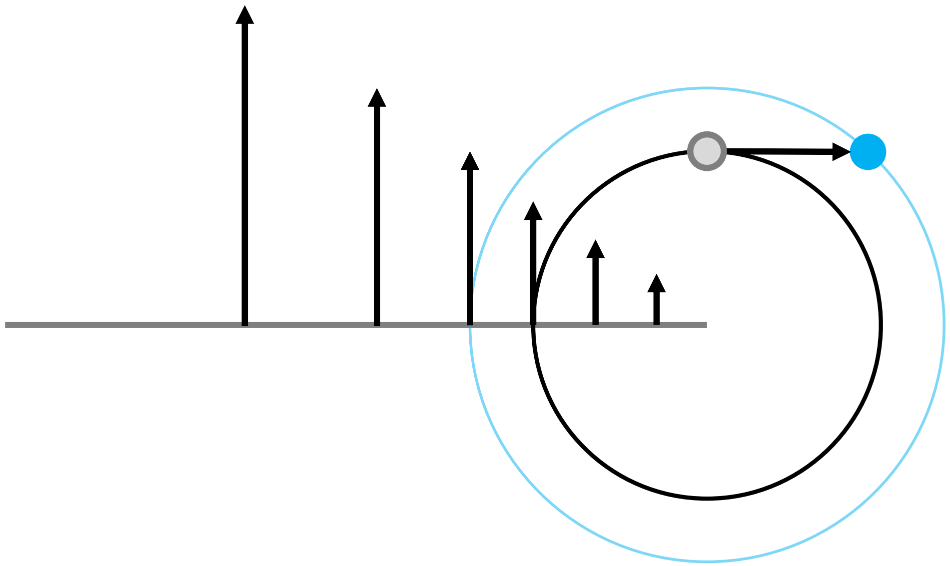
$h > \frac{2}{k}$: explode!

Stiff Equations

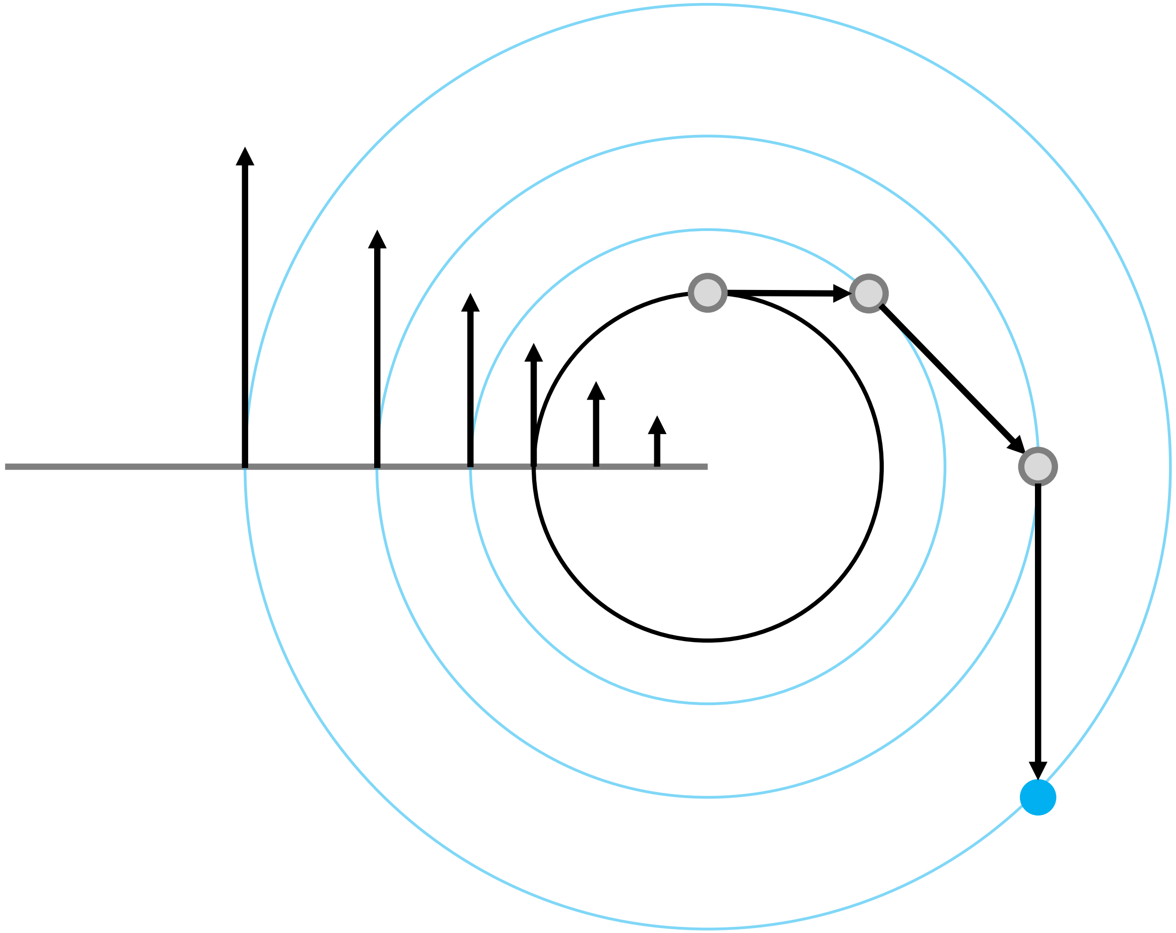
- In more complex systems, step size is limited by the ***largest*** k. In other words, one single stiff spring can screw it up for everyone else

- Systems that have some big k's mixed in are called ***stiff systems***

Explicit Euler Integration (a.k.a. Forward Euler)



Explicit Euler Integration (a.k.a. Forward Euler)



Implicit Euler Integration (a.k.a. Backward Euler)

Explicit Euler: $x(t+h) = x(t) + h\dot{x}(t)$

Implicit Euler: $x(t+h) = x(t) + h\dot{x}(t+h)$

Implicit Euler Integration (a.k.a. Backward Euler)

Explicit Euler: $$\underbrace{x(t+h)}_{\text{Uknown}} = \underbrace{x(t) + h\dot{x}(t)}_{\text{All known}}$$

Implicit Euler: $$\underbrace{x(t+h)}_{\text{Uknown}} = \underbrace{x(t)}_{\text{known}} + h\underbrace{\dot{x}(t+h)}_{\text{Uknown}}$$

Implicit Euler Integration (a.k.a. Backward Euler)

$$\underbrace{x(t+h)}_{\substack{\text{Uknown} \\ \text{(future)}}} = \underbrace{x(t)}_{\substack{\text{known} \\ \text{(present)}}} + h\underbrace{\dot{x}(t+h)}_{\substack{\text{Uknown} \\ \text{(future)}}}$$

- Interpretations
  i. We are looking for a future point $x(t+h)$ which exactly matches with your future velocity $\dot{x}(t+h)$

$$\dot{x}(t+h) = \frac{x(t+h) - x(t)}{h}$$

  ii. From the future point, if you take a Euler-step backward, then you should arrive at the current point

$$x(t) = x(t+h) - h\dot{x}(t+h)$$

An Example for $\dot{x} = -kx$

Explicit Euler:

$$x(t+h) = x(t) + h\dot{x}(t)$$

$$= x(t) - hkx(t)$$

$$= (1 - hk)x(t)$$

Implicit Euler:

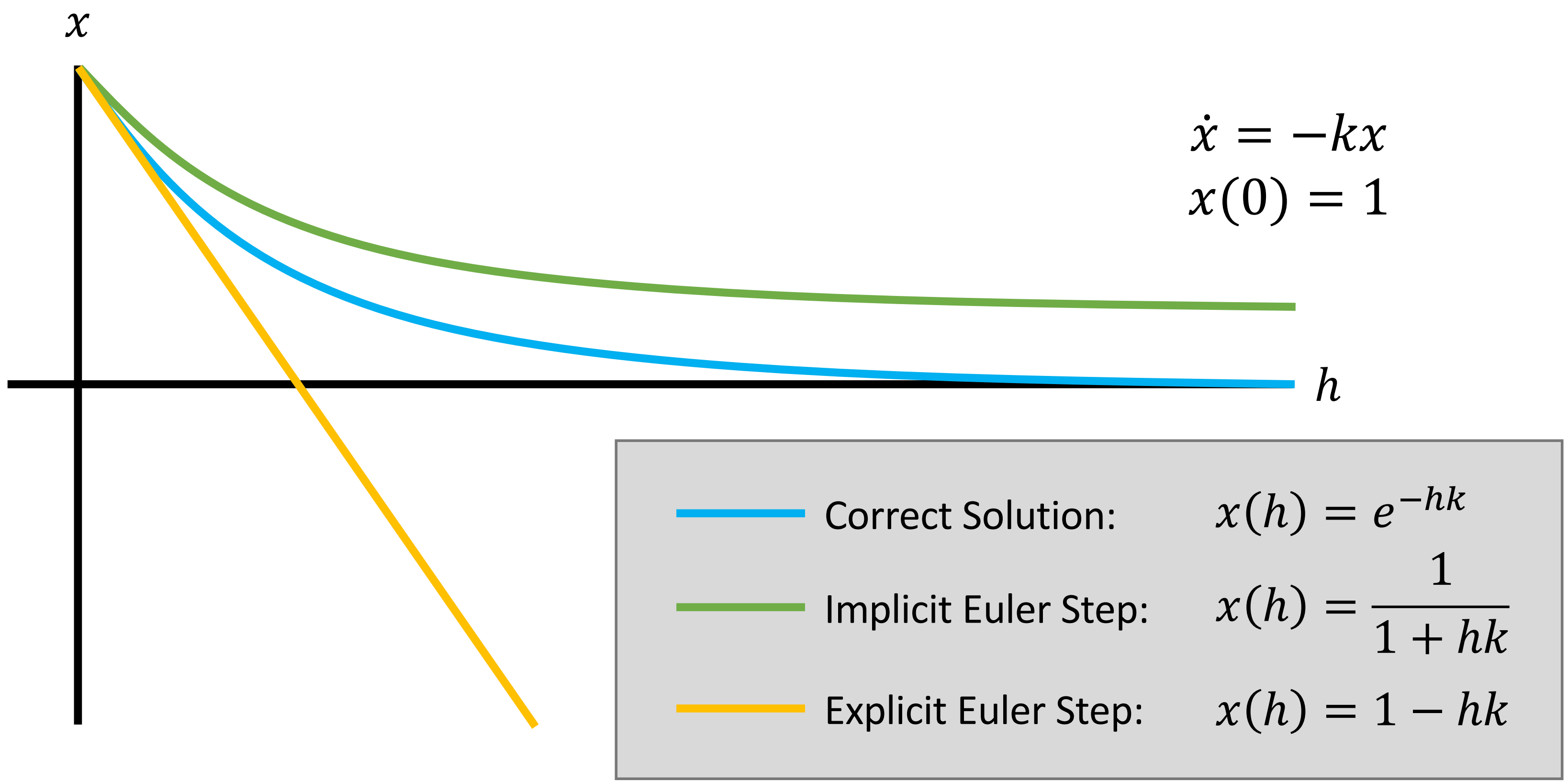
$$x(t+h) = x(t) + h\dot{x}(t+h)$$

$$= x(t) - hkx(t+h)$$

$$= \frac{x(t)}{1 + hk}$$

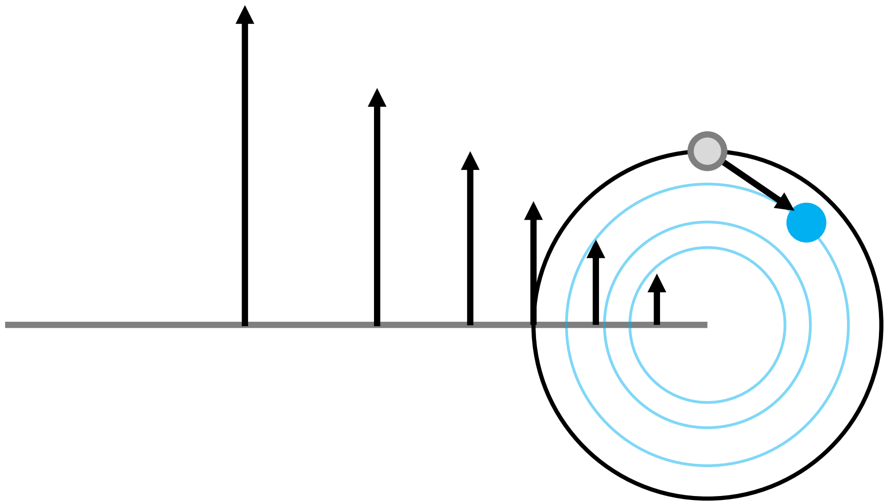
One Step: Implicit vs. Explicit

The graph tells us where we arrive after one step of integration

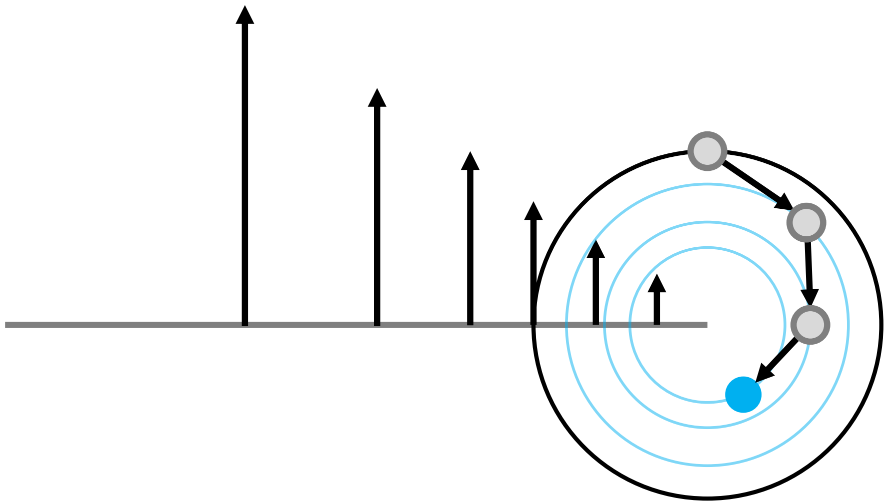


What would happen if $h \to \infty$?

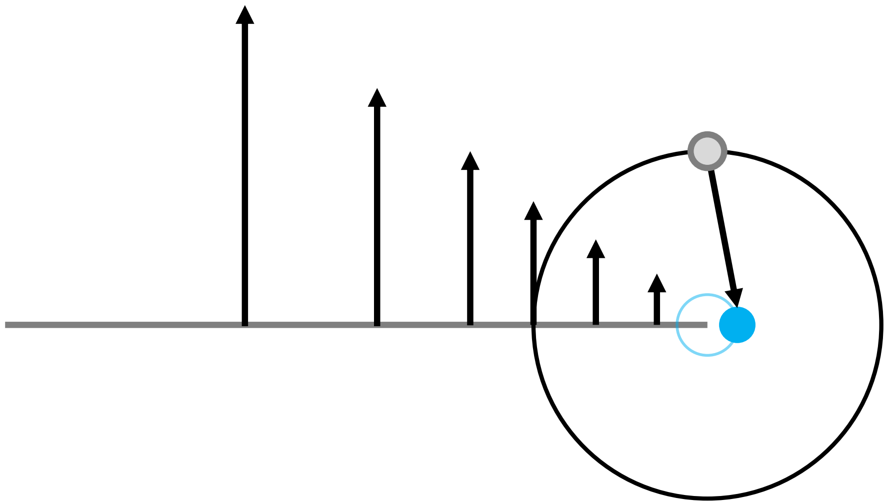
Implicit Euler Integration



Implicit Euler Integration



Implicit Euler Integration (Big Step)



It still shows stable behavior (i.e. not diverging)

Large Systems

- If f is a complex function, the equation for implicit Euler method becomes ***non-linear equation***, which is not trivial to find a solution

$$\mathbf{x}_{\mathrm{new}} = \mathbf{x}_0 + \mathrm{h}f(\mathbf{x}_{\mathrm{new}}) \quad \text{where } \dot{\mathbf{x}} = f(\mathbf{x})$$

- Our starategy is replace $f(\mathbf{x}_{\mathrm{new}})$ with a ***linear approximation*** based on f's Taylor series

Large Systems

$$\mathbf{x}_{\text{new}} = \mathbf{x}_0 + \mathrm{h} f(\mathbf{x}_{\text{new}})$$

$$\Delta\mathbf{x} = \mathbf{x}_{\text{new}} - \mathbf{x}_{\mathbf{0}}$$

$$= \mathrm{h} f(\mathbf{x}_{\text{new}}) = h f(\mathbf{x}_0 + \Delta\mathbf{x})$$

$$\boldsymbol{f}(\mathbf{x}_0 + \Delta\mathbf{x}) \approx f(\mathbf{x}_0) + \frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}} \Delta\mathbf{x}$$

$$= h\left(f(\mathbf{x}_0) + \frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}} \Delta\mathbf{x}\right)$$

$$\underbrace{\left(\mathrm{I} - h\frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}}\right)}_{\mathrm{A}} \underbrace{\Delta\mathbf{x}}_{\mathbf{u}} = \underbrace{h f(\mathbf{x}_0)}_{\mathbf{b}}$$

- A linear system is needed to be solved at each time step
- In many cases, f' will have a sparse structure (e.g. lattice structure)

Another Example

$$f\big(\mathbf{x}(t)\big) = \begin{pmatrix} -x(t) \\ -ky(t) \end{pmatrix} \qquad \mathbf{x}(\mathrm{t}) = \binom{x(t)}{y(t)}$$

Another Example

$$f(\mathbf{x}(t)) = \begin{pmatrix} -x(t) \\ -ky(t) \end{pmatrix} \qquad \mathbf{x}(\mathrm{t}) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

$$\frac{\partial f(\mathbf{x}(t))}{\partial \mathbf{x}} = \begin{pmatrix} -1 & 0 \\ 0 & -k \end{pmatrix}$$

$$\left(\mathrm{I} - h\frac{\partial f}{\partial \mathbf{x}}\right)\Delta\mathbf{x} = hf$$

$$\left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - h\begin{pmatrix} -1 & 0 \\ 0 & -k \end{pmatrix}\right)\Delta\mathbf{x} = h\begin{pmatrix} -x(t) \\ -ky(t) \end{pmatrix}$$

$$\Delta\mathbf{x} = -h\begin{pmatrix} 1+h & 0 \\ 0 & 1+hk \end{pmatrix}^{-1}\begin{pmatrix} x(t) \\ ky(t) \end{pmatrix} = -\begin{pmatrix} \frac{h}{1+h} & 0 \\ 0 & \frac{h}{1+hk} \end{pmatrix}\begin{pmatrix} x(t) \\ ky(t) \end{pmatrix} = -\begin{pmatrix} \frac{h}{1+h}x(t) \\ \frac{hk}{1+hk}y(t) \end{pmatrix}$$

$$\mathbf{x}(\mathrm{t}+\mathrm{h}) = \mathbf{x}(\mathrm{t}) + \Delta\mathbf{x} = \begin{pmatrix} \left(1 - \frac{h}{1+h}\right)x(t) \\ \left(1 - \frac{hk}{1+hk}\right)y(t) \end{pmatrix}$$

What would happen if $h \to \infty$?

Solving Second-Order Equations

- Most dynamics problems are written in terms of a second-order differential equation:

$$\ddot{\mathbf{x}} = f\big(\mathbf{x}(t), \dot{\mathbf{x}}(t)\big)$$

- This equation is easily converted to a first-order differential equation by adding a new variable $\dot{\mathbf{x}} = \mathbf{v}$

$$\frac{d}{dt}\begin{pmatrix}\mathbf{x}(t)\\ \mathbf{v}(t)\end{pmatrix} = \begin{pmatrix}\mathbf{v}(t)\\ f\big(\mathbf{x}(t), \mathbf{v}(t)\big)\end{pmatrix}$$

Solving Second-Order Equations

$$\Delta\mathbf{x} = \mathbf{x}(\mathrm{t}_0 + \mathrm{h}) - \mathbf{x}(\mathrm{t}_0)$$
$$\Delta\mathbf{v} = \mathbf{v}(\mathrm{t}_0 + \mathrm{h}) - \mathbf{v}(\mathrm{t}_0)$$

$$\begin{pmatrix}\Delta\mathbf{x}\\ \Delta\mathbf{v}\end{pmatrix} = h\begin{pmatrix}\mathbf{v}_0 + \Delta\mathbf{v}\\ f(\mathbf{x}_0 + \Delta\mathbf{x}, \mathbf{v}_0 + \Delta\mathbf{v})\end{pmatrix}$$
$$= h\begin{pmatrix}\mathbf{v}_0 + \Delta\mathbf{v}\\ \mathbf{f_0} + \frac{\partial f}{\partial\mathbf{x}}\Delta\mathbf{x} + \frac{\partial f}{\partial\mathbf{v}}\Delta\mathbf{v}\end{pmatrix}$$

$$\boldsymbol{f}(\mathbf{x}(t) + \Delta\mathbf{x}, \mathbf{v}(t) + \Delta\mathbf{v})$$
$$\approx \mathbf{f_0} + \frac{\partial \boldsymbol{f}}{\partial\mathbf{x}}\Delta\mathbf{x} + \frac{\partial \boldsymbol{f}}{\partial\mathbf{v}}\Delta\mathbf{v}$$

$$\Delta\mathbf{v} = h\left(\mathbf{f_0} + h\frac{\partial f}{\partial\mathbf{x}}(\mathbf{v}_0 + \Delta\mathbf{v}) + \frac{\partial f}{\partial\mathbf{v}}\Delta\mathbf{v}\right)$$

$$\underbrace{\left(\mathrm{I} - h\frac{\partial f}{\partial\mathbf{v}} - h^2\frac{\partial f}{\partial\mathbf{x}}\right)}_{\mathrm{A}}\underbrace{\Delta\mathbf{v}}_{\mathbf{u}} = \underbrace{h\left(\mathbf{f_0} + h\frac{\partial f}{\partial\mathbf{x}}\mathbf{v}_0\right)}_{\mathbf{b}}$$

Summary

- Implicit (backward) Euler integration requires to solve a linear equation at each timestep, which would take much time when compared to explicit Euler integration

- However, by doing so, our simulation becomes much more stable even with a larger timestep. This eventually compensates computational complexity required at each timestep in many cases

- There are some cases where computing $\frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}}$ is challenging